TOWARDS FULL INTEGRATION OF XML AND ADVANCED DATABASE CONCEPTS

David Olsen, Utah State University, 3515 Old Main Road, Logan, UT 84322, 435-797-2349, dolsen@b202.usu.edu
Bryan Marshall, Utah State University, 3515 Old Main Road, Logan, UT 84322, 435-797-2285, bryan.marshall@usu.edu
Richard Swart, Utah State University, 3515 Old Main Road, Logan, UT 84322, 435-797-8313, Richard.Swart@business.usu.edu
Vance Cooney, Eastern Washington University,

ABSTRACT

Most advanced database systems courses focus on core principles of database management. Given the ever increasing importance of web enabled databases generally but particularly the influence of XML, an alternative approach would be to teach the traditional core principles while integrating an XML module into the course. The focus of this paper is to elaborate how such an integration would be accomplished in an advanced database course. We have developed a module that introduces XML to students through a series of tutorials and assignment, and lays the foundation for subsequent courses in web services and web application programming.

INTRODUCTION

Advanced database courses traditionally cover in some depth the theory and concepts behind the design and implementation of relational databases. In recent years there has been mounting pressure to weave into this discussion content on integration with the Web as much e-commerce is facilitated by web-enabled databases. The purpose of this paper is to show how we enriched an advanced database class by adding an XML (eXtensible Markup Language, the emerging standard technology for database / web integration) module that extended a typical real world database example with an integrated XML component. To do this we developed a database using Microsoft's SQL Server 2000 and created tutorials and assignments to "pull" data from this database using the XML technology [4].

Prior to the development of XML there were efforts to adapt the precursor of HTML, SGML (Standard Generalized Markup Language) to accommodate the requirements of web–enabled database applications. SGML was a descendant of IBM's Generalized Markup Language (GML), originally developed in the 1960s to enable the sharing of machine-readable documents in large projects in government. It has had also been used extensively in the printing and publishing industries, but its complexity has had prevented its widespread application for small-scale general-purpose use [8]. When efforts to adapt SGML proved fruitless, Jon Bosak, Tim Bray, C. M. Sperberg-McQueen, and Jean Paoli of Microsoft, designed a simplified version of XML based on SGML that has since evolved into the standard XML that we have today [5].

The XML we use today is not merely an extension of HTML, it is a meta-language that can be used to define a language particular to a business domain and allow the exchange of data using this defined language. For example, instead of just the <H1></H1> tag pair that traditional HTML would offer that only has to do with how the text within tags is displayed, an XML tag has semantic value. Also, an XML tag example might be <Whsle_Unit_Cost></Whsle_Unit_Cost> and such a tag such as would

define the meaning of the data as well as the display format; i.e., data between the pairs would mean the wholesale unit cost of something, say a prescription drug. Furthermore, the organization receiving the XML file could easily import the data into their database systems. In short, XML allows a user to separate the presentation of data from its storage and management. This allows users a vast array of opportunities for using data and for using XML to exchange data with trading partners.

Since its inception, XML has been adopted by many developers as a way to describe data sets and their contents and to define how the data should be output or displayed on a web page (or, significantly, a cell phone, PDA or any of a number of other human readable or machine readable devices). Before XML, a client application accessing a database needed to translate the result set returned to a format that could be understood and displayed by a web application. XML removes the need for client side processing (given a XML compliant client) as the data and its formatting were are defined in the XML markup. The importance of XML is supported by the fact that Microsoft SQL Server 2000 supports XML, and the result sets can be returned directly in XML format, or data can be retrieved from an XML document as if it were a SQL Server table.

BRIEF LITERATURE REVIEW

After Codd's seminal 1970 paper on relational theory, hundreds of books and articles on relational databases have been published [3]. Knowledge of relational database design and normalization are staples of advanced database courses, and are key components of the IS2002 model curriculum and guidelines for undergraduate degree programs in information systems curriculum [2]. Likewise, many computer science departments rely strongly on the Computing Curriculum 2001 [6]. We believe that while model curricula such as IS2002 are extremely valuable guidelines, they are by their nature conservative, and likely to be slow in responding to emerging technologies such as XML. , but However, we believe that XML is going to be integral to future web enabled database architectures and that universities need to instruct students in XML. As a first pass at such development we have developed an XML Module that can be integrated into an advanced database management course. This will help students prepare to use modern development tools using XML, such as Microsoft's .NET framework, and Sun Microsystems's J2EE platform. What follows is a description of the teaching module we developed for advanced undergraduates and graduate students in an advanced database management course.

MOAB MEDICAL CLINIC – THE BACKGROUND CONTEXT FOR THE MODULE

One of the challenges of database instruction is creating interesting cases relevant to students. Many of Utah State University students are avid outdoor enthusiasts and our campus in located next to the entrance of a national forest with extensive biking trails. The biking trails around Moab are internationally famous, and we believe that students will easily identify with the nature of mountain biking and understand the common injuries from this sport. Many students have visited a clinic or emergency room for an injury at some point in their life, so a clinical model was chosen to illustrate principals of database design and to facilitate the student's comprehension of the queries that are assigned. Their familiarity with the material should assist them to interpret the correctness of the output of their queries and to formulate appropriate queries.

MMC Design - The Moab Medical Clinic (MMC) is located just outside of Moab, Utah near many of the famous slick rock mountain biking trails and off-road vehicle trails. It was primarily designed to handle emergency cases that result from accidents involving mountain bikes and off road vehicles. It

was located near many popular sites so that emergency response times could be minimized. Common cases at the clinic include fractured bones, concussions, abrasions, and puncture wounds from these outdoor accidents. Most of the patients entering the clinic will have common injuries and report pain.

The case approaches the design of the clinic's functions using a simplified model of care. Students are not expected to be familiar with medical clinic operations, and this teaching case builds upon common experiences in receiving medical care to determine the entities and relationships in the database.

Entity Relationship Modeling - Students are given a textual case based on the MMC. The students are to read the case and "ferret out" relevant database information. Students then determine the relevant entities and relationships based on information provided in the case. After the entities and relationships are determined, students create the database while enforcing the key constraints and business rules. A copy of the student assignment is located on the course website [4].

Entities - Students are given a textual case based on the MMC. The students are to read the case and "ferret out" relevant database information. Students then determine the relevant entities and relationships based on information provided in the case. After the entities and relationships are determined, students create the database while enforcing the key constraints and business rules.

Moab Medical Clinic XML Module - When the students finish creating and populating the database, they are ready to start learning about XML and how to apply it to this real world situation. The students will learn about the basic structure of XML, how to format XML on the web, the use of validation tools, and how to bring the data into SQL Server from an XML text file.

We include several tutorials and assignments designed to first teach the student the different methods of accomplishing a given task, and then the student uses this new knowledge to complete a correlating assignment. There are four tutorials and four assignments in the MMC XML module. The tutorials coincide with the assignments, teaching the students how to 1) create an XML document using SQL Server 2000; 2) attach an XSL-FO document; 3) attach a XSD document; and 4) bring a XML document into an existing SQL Server database. The tutorials can be found on the course website [4].

The first assignment involves creating the MMC database, as stated in the previous section. The rest of the assignments are interrelated and build upon each other. This is important because at the end of the XML module, the student should have a good understanding of how XML work. All of the assignments are found on the course website [4]. Because this module is being used in the advanced database course, the solutions are available upon request.

CONCLUSION

We have presented a comprehensive teaching case that demonstrates the integration of a XML module into an advanced database course. Based on a sound relational database built in earlier assignments, students proceed through a number of tutorials that develop their familiarity with key components of XML technology. We believe that this method offers a couple of advantages. First, students are reminded of the importance of good relational design principles as they build the MMC database into SQL Server 2000. Second, students learn XML technology in the context of an advanced database class. Third, this instructional approach lays an effective foundation for later courses in web services development or web application programming, without losing the course's focus on advanced database management.